IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

Programação Orientada por Objetos

Projeto

Vida no Campo

Programação II

2017

Índice

1. INTRODUÇÃO

O objetivo deste projeto é desenvolver, utilizando a linguagem Java e o paradigma de programação orientada a objetos, uma aplicação que permita aplicar pragmaticamente soluções para os conceitos aprendidos no decorrer da unidade curricular, os quais permitem satisfazer um elevado e importante número de princípios fundamentais da Engenharia de Software, nomeadamente: a modularidade conceptual e de implementação, a correção de erros e eficiência, e, ainda, a manutenção e a extensão de uma aplicações.

Os conceitos fundamentais da programação orientadas aos objetos está bastante alicerçada na solução que se pretende que responda as necessidades do projeto, nomeadamente os conceitos de classe, objeto, abstração de dados, proteção e encapsulamento. Para além disto está presente os conceitos avançados tais como o polimorfismo, hierarquia, herança e agregação.

Para tal pretende-se que os alunos desenvolvam uma aplicação que funcione num computador onde um programador poderá realizar simulações ou então um utilizador poderá correr simulações.

O projeto será desenvolvido em duas fases. A primeira fase consiste na modelação e implementação do conjunto de classes que permitam representar a lógica da aplicação; a segunda fase será dedicada à criação da interface em modo consola ou gráfica para interagir com o utilizador.

Tenha particular atenção ao uso do paradigma de POO na modelação das classes, i.e., ao correto uso dos conceitos de encapsulamento, herança, classes abstratas, polimorfismo, interfaces, maximização da coesão (responsabilidade única), minimização do acoplamento, desenho orientado por responsabilidades, etc. Uma modelação bem pensada facilitará todo o processo de desenvolvimento e manutenção. Por fim a utilização do mecanismo de exceções de JAVA permitirá implementar os tão importantes aspetos da segurança e robustez do software, no qual o JAVA, permite obter todo o potencial do muito claro e limpo mecanismo de tratamento de exceções.

2. REGRAS DO JOGO

Numa quinta estão vários animais. Certos animais são domésticos: cães e gatos. Os animais vivem com liberdade relativa, mas confinada ao espaço da quinta. Quando os animais interagem ocorre alguns problemas de que pode resultar em arrufos (entre gatos e cães) ou mesmo a perda da vida (entre presas e predadores).

Alguns animais, vivem na quinta, perto da casa, mas não são considerados animais domésticos, são eles os pássaros e os ratos. Estes animais também se deslocam pela propriedade, soltos, mas nem sempre livremente por causa dos predadores.

Alguns comportamentos são partilhados por diferentes animais. Alguns comportamentos são distintos de um único animal. Por exemplo voar é algo intrínseco aos pássaros.

2.1.CARACTERÍSTICAS DOS ANIMAIS

Neste mundo os animais podem ser predadores ou presas. Um predador (gato ou cão) é descrito por capturar e comer uma presa. Quando isto acontece, consome toda a sua energia.

Todos os animais podem correr, mas somente os pássaros podem voar. Quando correm ou voam consomem alguma energia para poder concretizar. Quando a energia termina, não podem correr mais e têm de dormir para recuperar forças.

Contuso, além do repouso, os cães e os gatos, como são predadores, podem recuperar energia comendo ratos. Um rato devorado perde a energia, toda a energia. Esta é transferida para o predador. Os gatos, por serem ágeis, também conseguem comer pássaros.

Por vezes, os cães ficam incomodados e perdem a paciência e, por isto, atacam os gatos. Ambos perdem energia no processo. Na iteração entre animas temos os seguintes comportamentos:

1. O gato quando tenta comer o pássaro corre na sua direção. O pássaro voa. O gato consegue, de acordo com uma taxa de sucesso, por vezes capturar o pássaro em voo e quando isto acontece a energia do pássaro passa para o gato. Neste processo o gato perde energia a correr e o pássaro a voar. Quando o pássaro escapa a sua energia aumenta de acordo com um valor definido;

2. O cão e o gato quando tentam comer o rato correm na sua direção. O rato corre também. Por vezes, de acordo com uma taxa de sucesso, o rato é comido. Neste caso a sua energia passa para o animal que o comeu. Quando o rato escapa a sua energia aumenta de acordo com um valor definido.
3. Um cão consegue sempre atacar um gato. Quando isto acontece a sua energia diminui num valor pré-determinado. O gato também perde energia pelo facto de ter sido atacado pelo cão, também num valor pré-determinado. Neste cenário a energia que estes animais perdem é diferente (i.e. é menor) da quantidade de energia que perdem quando tentam comer um rato ou um pássaro.

As presas são caracterizadas por terem, além da função de consumo de energia, uma função que indica se escaparam ao predador.

2.2.O QUE DEVE SER MODELADO

Modele, pelo menos os conceitos "cão", "gato", "pássaro" e "rato". Além da energia, os cães e os gatos têm nome. Por defeito existe um conjunto de valores para a energia ganha ou perdida. Estes valores podem ser alterados ou durante a criação dos animais. Deve pois existir dois tipos de construtores.

Para um predador comer uma presa tem de a perseguir para a capturar (podendo a perseguição ser ou não bem-sucedida). Um cão consegue capturar um rato em cada 25 tentativas. Para os gatos, o rácio é 1 em 5 (ratos) e 1 em 10 (pássaros). A perseguição consome a mesma energia que correr (para cada interveniente), mas a presa recebe um bónus de 5 unidades se escapar. Se a presa estiver a dormir, é apanhada 1 em cada 2 tentativas.

Os cães têm por defeito a energia de 10000 e gastam a energia de 50 a correr. Quando atacam um gato gastam a energia de 100, por sua vez o gato perde a energia de 25.

Os gatos têm por defeito a energia de 500 e gastam a energia de 25 a correr.

Os ratos têm por defeito a energia de 50 e gastam a energia de 2 a correr.

Os pássaros têm por defeito a energia de 20 e gastam a energia de 5 a correr e 2 a voar.

3. SUGESTÕES DE DESENVOLVIMENTO

Inicie por reconhecer as categorias que estão presentes no problema: estas irão corresponder às classes a implementar.

Determine eventuais hierarquias. A definição de uma hierarquia adequada é um passo importante para o sucesso da implementação

Depois identifique as relações entre as diversas entidades envolvidas, ponderando as vantagens e desvantagens das diferentes alternativas.

Para cada classe identifique a informação que será necessária armazenar para caracterizar o estado dos seus objetos (serão os atributos). Em continuação, para conseguir as manipulações base da informação, crie os respetivos construtores, seletores e modificadores, que façam sentido no contexto.

Finalmente implemente, nas classes próprias, métodos específicos do problema, no caso presente os relacionados com as regras e funcionamento da simulação.

Tenha em atenção a implementação adequada dos construtores, modificadores e seletores, de modo a não existirem estados instáveis nos objetos.

4. SIMULAÇÕES

Construa as classes fundamentais e um programa que permitir simular a iteração com o utilizador. A iteração pode ser em modo consola ou através de interface gráfica. A iteração deve permitir a codificação da criação dos animais, corridas, ataques, etc. Apresente o estado inicial dos animais e o estado final (depois de algumas interações). Pode também, numa iteração mais elaborada, através de menus ou de uma interface gráfica, permitir o utilizador escolher as ações que pretende realizar deve fazer (por exemplo criar animal, executar comportamento.

Em ambos os cenários, os gatos correm, perseguem pássaros e ratos e são atacados pelos cães, que também podem correr e perseguir e comer ratos. Os animais dormem automaticamente se ficam sem energia (exceto quando são devorados: nesse caso devem ser considerados mortos).

5. CODIFICAÇÃO

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação PascalCase para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize o símbolo '_' nos identificadores (exceto nas constantes), nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

5.1. CONSTITUIÇÃO DE GRUPOS

Cada projeto deverá ser elaborado individualmente.

6. ENTREGA DO PROJETO

O projeto deverá ser entregue até à data limite especificada por via exclusivamente eletrónica utilizando a área dos trabalhos no Moodle. Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP.

Não serão aceites quaisquer projetos entregues fora do prazo!

Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- O Manual Técnico onde conste uma breve descrição do programa, incluindo a explicação das classes/interfaces implementadas, principais atributos e métodos e suas relações;

- O código fonte do programa na forma de projeto em *NetBeans*, com um ou mais *main* de testes a funcionar e com todas as funcionalidades implementadas. Num dos main deve ser implementado de uma interface com o utilizador em modo de consola.
- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura: <*numAluno1*>..zip.

7. REGRAS DE AVALIAÇÃO

A avaliação do projeto está sujeita às seguintes regras:

- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da Programação Orientada por Objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento da linguagem Java.
- Serão premiadas a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- Para a avaliação oral irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão anulados.

8. CRITÉRIOS DE AVALIAÇÃO

Funcionalidades	**35**
Atributos do gato	5
Atributos do rato	5
Atributos do cão	5
Atributos do pássaro	5
Comer o rato pelo gato	5
Comer o rato pelo cão	5
Atacar gato	5
Implementação	**40**
Estrutura de classes	15
Conhecimento e boa utilização da linguagem	15
Bom estilo (nomes, comentários, indentação)	10
Simulação por consola/GUI	**25**
Ecrãs de criação de animais	**5**
Ecrãs de comer o rato e o pássaro	**5**
Ecrãs de atacar o gato	**5**
Ecrãs para fazer com que animais possam dormir/acordar	**5**
Ecrãs para visualizar estado da quinta	**5**

9. RESUMO DAS DATAS IMPORTANTES

1. Entrega do Projeto
   A entrega do projeto será até **às 23:00, dia 19 de Janeiro de 2018 via Moodle.**

2. Avaliações do projeto
   As avaliações serão realizadas nos dias da semana de **22 - 26 de janeiro de 2018**